Installation of Utah's pcc-20 and runtime system. Last updated 4/25/83.

Make sure that you have installed the patch to MACRO described in SPR 20-15625, 10-MACRO-261/1212, dated 15-APR-81. It concerns mis-conversion of numbers in decimal radix (for example, the largest neg number gets converted to 0, and that number is FHSLF == 0400000,,0 !). This was NOT fixed in the Release 5 distribution.

Logical name SCRATCH: must point to a directory suitable for temporary files. This can be either system-wide if you have such a directory, or each pcc user can define his or her own. It's definitely best to make the directories <pcc> & <unix>, but this isn't absolutely necessary at this point. [I.e., if pcc: points to something other than <pcc>, then #include's of the form <sys/foo.h> will break. If <unix.lib> does not exist, then any program using termlib will break, as well as any other program referencing a file currently on <unix.lib>.] It's also best to define the system wide logical names pcc: and unix: pointing to them, but again it's not strictly necessary if you use the PCL commands.

==========
There are 5 savesets on a full source tape, while on a binary tape there are only four, somewhat smaller ones. You should be able to run successfully with just the first two savesets loaded. The number of pages given here is an approximation only.

*ssname Utah pcc-20 Documentation
Both source and binary-only tapes: <pcc.doc> approx 25 pages

*ssname Utah pcc-20 Executables and Libraries
Both: <pcc>,<pcc.sys>,<unix>,<unix.lib>,<unix.man> 1900 pages full
1100 pages binary

*ssname Utah pcc-20 Preprocessor and compiler sources
Source tape only: <lepreau.cpp.dist> 170 pages

*ssname Utah pcc-20 Runtime system sources
Src: <lepreau.3>,<lepreau.3.mod>,<lepreau.3.local>,<lepreau.support> 440 pages
Binary: <lepreau.3.mod>,<lepreau.3.local>,<lepreau.support> 220 pages

*ssname Utah pcc-20 Couple useful programs sources
Both: <unix.src> 23 pages
==========

Rename the files to the proper directories if you've loaded them elsewhere. Set the protection of these .exe files in the <pcc> directory to 777712: cpp, c1, c2.

If you aren't running CMU's PCL Exec, using the compiler (at this point) will be much less pleasant. If you don't have it, look at the MIC files in the <pcc> directory. They will help, but they could use some work. If you don't have PCL and decide to whip up a driver cc program, that would be lovely and please let me know of it so it can be distributed to others. It shouldn't be hard; just little motivation for it here. I don't recommend using the library fork/exec calls for this as the former is currently way to inefficient.

FOR PCL USERS <
[[The file pcc:init.cmd contains the variable 'tempdir' pointing to SCRATCH: as above. If you choose to change the compiler (code.c) to put junk some other place than this you will want to change it here too. The same variable is also in pcc:pcc.pcl, but it is not essential to alter it here... Look for the string "INSTALLATION DEPENDENT".]]

If you are running Rel 5 and "old" PCL (and maybe even if not Rel 5, but the problem wasn't nearly as bad in 4) there's an exec patch which you should install. It is not essential, but w/o it jobs will sometimes hang if a PCL cmd is ^C'ed at the "wrong" moment. It is described in the file <pcc.doc>pcl-exec.patch. ("Old" PCL is defined as anything before CMU's "recent" (approx Nov 82) semi-formal release. The problem may conceivably still exist in the "new" one, but I doubt it.)

Look at pcc:[45]install-pcl.mic. You may not need to run it if you have nearly the same version of the exec as we do [5(712)], but it's probably a good idea. Then run the appropriate one via "mic pcc:4install-pcl" (or 5).
NOTE: we haven't run under Rel 4 for quite some time, so the PCL I have here might no longer work.
>

Make any necessary changes to the file <pcc.doc>pcc.helpfile and copy it to hlp:pcc.hlp.

Run a test or two, such as compiling, linking & running asmlist.c on itself. See <pcc.doc>asmlist.doc for how to use it.

<unix.lib>termcap must have the terminal identifiers you use upper-cased; otherwise no termlib pgms will work. I have only done it for a couple of them that we have here, e.g. "T10". You should also move your local terminals to the front, as that will make termlib programs start up much faster.

Announce it; <pcc.doc>initial-bboard.msg is suitable for a bboard message.

****
Whoever is maintaining this stuff locally, please let me know who you are and how I can contact you so that I can keep you informed of bug fixes, enhancements, etc. A net address (arpa, uucp, csnet) is MUCH preferred if at all possible. My address is below.
****

WARNING: all this stuff is provided as-is, warts and all. It is expressly a "version," not a distribution, so don't expect the world. In particular, detailed documentation of the current limitations of the library routines is sometimes absent, thought the file <pcc.doc>current.status is fairly up to date. The final authority is the source, as usual. Expect another version sometime with major compiler improvements. Meanwhile, I may send out updates via the Arpa or Uucp nets. If you aren't on one of them, get on Uucp if you have a Unix system. (Note that University of Washington and Rutgers University are both running uucp on their 20's with moderate success, so you may be able to set up a uucp link straight from you 20.)

About the monsym-derived files: the excerpted ones in the library source dirs (of form mon_foo.h) exist because cpp can't handle the full one. Should make a huge version of cpp, or better pre-load a cpp with monsym. Anyway, some of them are in an obsolete format and some are in new format matching <pcc>monsym.h. The latter may blow up occasional library modules which expect the old format and never got re-compiled.

For installed use, it probably would be a good idea to do a Tops-20 MAKLIB on the library and remove the multiple search from clink.pcl, or clink.mic. (I haven't done it because I'm actively working on it and it changes constantly.)

If you need to modify or recompile the library, preprocessor, or compiler see <pcc.doc>regen-system.notes.

Good luck!

Contact me about problems:
Jay Lepreau
Computer Science Dept.
Univ. of Utah
Salt Lake City, Utah 84112
801-581-4285, 582-7214

bug-pcc@Utah-20, harpo!utah-cs!bug-pcc (which is me right now)
lepreau@Utah-20, decvax!harpo!utah-cs!lepreau

CORRECTION TO <PCC.DOC>CURRENT.STATUS

A slightly obsolete version of current.status is on the 4/25/83 tape--
corrections to three of the entries are fairly important. Please modify your
CURRENT.STATUS file with these three.

open, close — additional modes supported: append, exclusive open, arbitrary byte size, default (existing) byte size. See <file.h> for optional 3rd param. Opens currently default to 7 bit.

signal, kill, alarm — Crude, preliminary implementation of signal. Has known bug when input "^C\r" in CBREAK mode. SIGQUIT maps to ^B, not ^\.
These are supported to varying degrees:
(hup),int,quit,ill,(kill),fpe,(bus,segv),alrm,term

fopen, freopen, fdopen — read/write, append, "binary" modes also supported via Berkeley + convention, 'a' == append, 'b' = binary, e.g.
fopen(foo, "r+"); open(foo, "wb"); fopen(bar, "a");